REMARKS/ARGUMENTS

Applicant thanks the Examiner for the interview of September 15, 2006 in which the Examiner and Applicants counsel discussed a proposed Amendment and Response previously forwarded to the Examiner for review. The Amendment and Response is substantially identical to this paper. During the interview the Examiner and Applicants counsel discussed the insertion of the word "direct" in the claims. Specifically, Jordan, et al. teaches tracking the frequency of forwarded requests but not direct requests. The Examiner further indicated that dependent Claim 47 and other claims containing similar limitations may be allowable over the art. However, the Examiner indicated that further searches were required before such a decision could be rendered. No agreement was made as to the allowability of the subject application.

The Examiner rejects claims 40, 43-46, 48, 69, and 72 under 35 U.S.C.§103(a) as being unpatentable over Jordan, et al. (U.S. 6,438,652) in view of Applicant's Admitted Prior Art (AAPA); claims 47 and 73 under Section 103(a) as being unpatentable over Jordan, et al., in view of AAPA and further in view of Balijepalli, et al.; claims 41-42, 49-53, 55-56, 70-71, 74, and 76-79 under Section 103(a) as being unpatentable over Jordan, et al., in view of AAPA and further in view of Wallace, Jr.; claim 81 under Section 103(a) as being unpatentable over Jordan, et al., in view of AAPA further in view of Wallace, Jr., as applied to claim 76 and further in view of Balijepalli, et al.; claim 61 under Section 103(a) as being unpatentable over Jordan, et al., in view of Balijepalli, et al.; claims 63-67 and 83-87 under Section 103(a) as being unpatentable over Jordan, et al., in view of Wallace, Jr.; and claims 54, 57-60, 62, 75, 80, 82, and 88 under 35 U.S.C.§102(e) as being anticipated by Jordan, et al.

Applicant respectfully traverses the Examiner's rejections. The cited references fail to teach or suggest at least the following italicized features of the pending independent claims:

40. An arrangement for serving information requests, comprising:

a plurality of information servers connected to a communications network, all of the information servers having a common address on the communications network and serving a set of information to clients, each of the information servers being configured to receive a transaction request associated with an individual transaction and to provide a response to each transaction request; and

a content director connecting the information servers to the communications network and distributing transaction requests among the information servers comprising:

a flow switch that selects an appropriate information server to service each transaction request and thereafter forwards at least portions of the transaction request to a selected one of the information servers;

a cache processor that accesses a cache;

the cache that stores, in a hot locator table identifying information frequently requested from the information servers and including, for each locator identifying corresponding information, a hit counter indicating a number of direct transaction requests, received by the plurality of servers over a determined time interval, requesting the corresponding information;

a digest generator that generates, when the hit counter for a locator indicates at least a threshold transaction request receipt frequency but not when the hit counter fails to indicate at least a threshold transaction receipt frequency, a digest value pointing to the location in the hot locator table where the corresponding entry is stored; and

a digest store that stores the digests corresponding to frequently requested content.

54. In an arrangement comprising a plurality of information servers connected to a communications network, each of the information servers being configured to receive a transaction request associated with an individual transaction and to provide a response to each transaction request, a method for serving transaction requests from clients comprising:

maintaining a hot locator table identifying information frequently requested from the information servers, the hot locator table including, for each locator identifying corresponding information, a hit counter indicating a number of direct transaction requests, received by the plurality of information servers over a determined time interval, requesting the corresponding information;

generating, when the hit counter for a selected locator indicates at least a threshold transaction request receipt frequency, a digest value pointing to the location in the hot table where the entry corresponding to the selected locator is stored; and

accessing a digest store comprising the digest values to select an information server to service a transaction request for frequently requested information.

69. An arrangement for serving information requests, comprising:

a plurality of information servers connected to a communications network, all of the information servers having a common address on the communications network and serving a set of information to clients, each of the information servers being configured to receive a transaction request associated with an individual transaction and to provide a response to each transaction request; and

content director means for connecting the information servers to the communications network and distributing transaction requests among the information servers comprising:

first flow switching means for parsing plain text transaction requests to locate selected fields, selecting an appropriate information server to service each transaction request, and thereafter forwarding at least portions of the parsed transaction requests to a selected one of the information servers;

cache processing means for accessing a plurality of objects in response to communications received from the first flow switching means;

cache means for storing, in a hot locator table, the plurality of objects including locators identifying information frequently requested from the information servers and including, for each locator identifying corresponding information, a hit counter indicating a number of direct transaction requests, received by the plurality of information servers over a determined time interval, requesting the corresponding information;

digest generator means for generating, when the hit counter for a locator indicates at least a threshold transaction request receipt frequency but not when the hit counter fails to indicate at least a threshold transaction receipt frequency, a digest value pointing to the location in the hot locator table where the corresponding entry is stored; and

digest store means for storing the digests corresponding to frequently requested content.

75. In an arrangement comprising a plurality of information servers connected to a communications network, each of the information servers being configured to receive a transaction request associated with an individual transaction and to provide a response to each transaction request, a method for serving transaction requests from clients comprising:

maintaining a hot locator table identifying information frequently requested from the information servers, the hot locator table including, for each locator identifying corresponding information, a hit counter indicating a number of direct transaction

> *requests, received by the plurality of servers over a determined time interval, requesting the corresponding information;*
>
> *when the hit counter for an locator indicates at least a threshold transaction request receipt frequency, locating the information associated with the locator at a cache information server and thereafter directing a transaction request for information associated with the locator to a cache information server; and*
>
> *when the hit counter for a locator falls below a threshold transaction request receipt frequency, directing a transaction request for information associated with the locator to an origin information server.*

Jordan, et al.

Jordan, et al., is directed to a server farm including cooperating cache servers in which information requests are forwarded to a cooperating cache server if the requested object cannot be found locally. As a consequence of direct and forwarded information requests, cache servers can become overloaded. Load balancing is effected by a load monitor 120 that shifts one or more of the forwarded requests for the object between cooperating cache servers based on a load condition and a forwarding frequency for the object. The load condition is a weighted sum of a count of the forwarded requests and a count of direct requests to the cooperating cache server. Overloaded servers are those with loads exceeding a threshold.

More specifically, Jordan, et al., states:

> FIG. 3 shows an example of a logic flow for steps taken by the load monitor 120 in response to a request 125 from a cache server 150 because of a cache miss. As depicted, in step 201, it checks to see if the requested object/partition can be found in the caching table. If not, in step,202, a new entry is created for the object/partition and a cache server is assigned as its owner. After the entry is located in the caching table, in step 203, *the forwarding frequency 1011 is updated, e.g., incremented by 1*. The load monitor then examines the load table 102 *to see if the owner is currently overloaded (and that the forwarding frequency 1011 is a significant contributor thereto*), in step 204. If yes, in step 205, the load monitor finds an *underloaded* (or less loaded) cache server and assign it as the new 10122 (or shared) owner 10122 of the requested object. The

> ownership information 1012 for the object in the caching table 101 is updated accordingly . . . Conversely, if a shared ownership model is used, in step 208, when the load condition 10211 and *forwarding frequency* 10111 for a shared ownership object (p 10101) *drops below a predetermined threshold*, in step 209, the shared ownership (B, A 10121) can be merged to a single ownership and one of the copies purged from one of the cache servers A 10121, e.g., to make room for another hot object.
>
> . . .
>
> From the caching table 101, it can be seen that object p 10101 is not cached on server C, but it is cached on ("owned" by) cache server B (assuming B, A 10121 is initially only solely designated by B). In response to a cache miss on object p, server C 10223 sends a request to the load monitor 120 for object p. *Depending on the load condition 10212 and forwarding frequency 1011* of requests for p 10101 on server B, the load monitor may forward the request to server B, asking it to send a copy of object p to server C. Or, if server B is currently overloaded or is trending as such, the load monitor might shift the forwarded request by finding an underloaded (or less loaded) server to serve as a new (or shared as in B, A 10121) owner of object p. The request is then forwarded to the new (or shared e.g., A) owning server for the object.
>
> . . .
>
> Another alternative is a load balancing method for a collection of cooperating proxy cache servers where a hash function is used to locate a copy of a locally missed object. In this case, the object space can be partitioned among the cooperating proxy cache servers 150, with one partition for each cache server. In order to achieve load balancing by shifting *forwarded requests*, one can change the hash function so that *forwarded* requests will not go to overloaded servers. One preferred approach is to hash the object space into a large number of buckets, much larger than the total number of proxy cache servers. These hash buckets are then assigned to the cache servers, with the goal of balancing the loads among them. Periodically, one can move one or more hash buckets from one overloaded server to an underloaded server, effectively changing the hash function.

(Col. 6, line 50, to col. 7, line 52 (emphasis supplied).)

Jordan, et al., is distinguishable from the present invention for a number of reasons. First because Jordan, et al., is directed to monitoring and responding to the load condition of each cooperating cache server *individually,* the load condition is determined on a server-by-server basis and *not* for the servers as a collective group. Second, Jordan, et al., monitors the request frequency not for specific locators, which are associated with specific information, but rather for specific cache servers, each of which contains information associated with a number of different locators. Third, Jordan, et al., does not measure hotness of content based on *direct* requests. It only monitors the frequency of forwarded requests for specific content. There is thus no motivation or suggestion to modify Jordan to monitor the load condition for the group of cooperating cache servers because Jordan is specifically directed to load balancing caused by requests forwarded by one cache server to another cache server. Monitoring load conditions for the group of servers would fail to recognize the effects of forwarded requests.

Jordan, et al., fails to teach or suggest the italicized features above. Depending on the independent claim, these features include: (a) the maintenance of a "hot locator table" based not on the number and/or frequency of requests for *specific objects* from a selected server but on the number and/or frequency of requests *received by a group of servers* for the objects; (b) the generation, *when a hit counter for an indicates at least a threshold transaction request receipt frequency but not when the hit counter fails to indicate at least a threshold transaction receipt frequency*, of a digest value pointing to the location in the hot table where the corresponding entry is stored; and/or (c) a digest store that stores the digests corresponding to frequently requested content.

The Examiner responds to these arguments by relying on Figs. 2a and 2b and col. 1, line 61 through col. 2, line 3, col. 6, lines 6-13; col. 4, lines 24-37; and col. 6, lines 42-45, and states:

> As such, Jordan teaches the features "(b) the generation, when a hit counter to an indicates at least a threshold transaction request receipt frequency, of a digest value pointing to the location in the hot table where the corresponding entry is stored; and"or (c) a digest store that stores the digests corresponding to frequently requested content.

(Office Action at page 5.)

A careful review of this language in Jordan, however, fails to support the Examiner's position. Jordan clearly teaches load balancing on servers but not on content. Fig. 2b is the alleged equivalent of the hot URL table. However, the cache server, and not its content, is identified in column 1022. Fig. 2a, the caching table, does teach object id/partition id 1010 indexed against *forwarding* frequency 1011 and ownership 1012. The caching table however does not track the frequency or direct content requests for specific information and is not limited to hot content and includes all content (see forwarding frequency of "2" for server A, "25, 2" for servers B and A, respectively, and "10, 50" for servers C and A, respectively).

For this reason, the Examiner's reliance on the abstract is misplaced. The Abstract states "[a]n overload condition is detected if for example, due to reference skew, some objects are in high demand by all the clients and the cache servers that contain those hot objects become overloaded due to forwarded requests." Although unquoted by the Examiner, the Abstract further states "the load is balanced by shifting some or all of the *forwarded* requests from an overloaded cache server to a less loaded one." (Emphasis supplied.) Thus, as noted by Applicant Jordan, et al., clearly teaches load balancing on servers but not on content.

The Examiner further relies on the use in Jordan, et al., of forwarding frequency and its reference to hashing as teaching "the generation, when a hit counter for an invariant indicates at least a threshold transaction request receipt frequency, of a digest value pointing to the location in the hot invariant table where the corresponding entry is stored". This statement disregards the facts that the direct request frequency is not tracked by Jordan, et al., and that the hashing is performed even when the forwarding frequency is less than the threshold transaction request receipt frequency.

These deficiencies of Jordan, et al., are not overcome by the other cited references.

Wallace, Jr

Wallace, Jr., is directed to establishing secure network user states. The states are secured by creating a first key from a received user key, encrypting user data with the key, storing the encrypted user data in a cookie, and sending the cookie to the computer, such that subsequently a secure user state between the server and user is established by receiving the cookie and the user key from the computer. The computer in response creates a second key matching the first key, decrypts, using the second key, encrypted user data extracted from the cookie, and establishes the secure state based on the decrypted user data. The cookie contains user-specific information, such as the user's identity, when the user has visited a selected website, user's IP address, mailing address, email address, age, sex, credit card information, login/password combinations, preferences, hobbies, education level, browsing (click) history, browsing history with click frequency, browsing preferences, assigned primary keys, assigned GUIDs, etc. (Page 3, ¶0054.)

Balijepalli

Balijepalli, et al. is directed to a web-based multi-user system and method for identifying, retrieving, and delivering information corresponding to items contained in a user search list from two or more information sources on the World Wide Web. The system includes a central server that periodically searches the information sources on the web for information corresponding to the items contained in the user search list. The central server retrieves the information onto a storage database where a determination is made as to whether the information is current. The central server electronically delivers only the current information to the user. The system and method therefore provides to the user automatic and periodic electronic reports containing only current or updated information corresponding to items contained in the user search list that the user desires to track.

Accordingly, the pending independent claims are allowable.

The dependent claims provide further reasons for allowance.

By way of example, dependent claim 41 requires the flow switch to parse plain text transaction requests to locate selected fields and a cryptographic module in the flow switch to decrypt, prior to parsing and information server selection by the flow switch, cipher text transaction requests and provide plain text transaction requests to the flow switch, wherein, prior to decryption, the cipher text transaction requests have not been routed by another flow switch. (See also claims 55, 70, and 76). Although Wallace, Jr does teach parsing an encrypted payload, Wallace, Jr fails to teach or suggest parsing *plain text* transaction requests prior to routing of the request by the flow switch.

Dependent claim 42 is directed to the receipt of first and second encrypted transaction requests from different clients having a common electronic address and served substantially simultaneously by different information servers, wherein at least some of the responses include a cookie, wherein the cookie is generated by the information server previously assigned by the flow switch to service transaction requests from the client, and wherein the flow switch uses at least one of a locator, a cookie, and a tag in the parsed plain text equivalent of each transaction request to select an appropriate information server to service each of the first and second transaction requests. (See also claims 56, 71, and 77.) Dependent claim 43 requires each locator in the hot locator table to have a corresponding timestamp indicating when the respective entry was last updated, and a tag identifying a corresponding information server providing the corresponding information. (See also claim 57 and 80). Dependent claim 50 requires the flow switch to tag a transaction response, the tag identifying an information server generating the transaction response. (See also claims 64 and 84.) Dependent claim 51 requires at least some of the responses to include a cookie, wherein the cookie is generated by the information server previously assigned by the flow switch to service transaction requests from the client, and wherein the cookie is different from the tag. (See also claims 65, 74, and 85.) Dependent claim 52 requires the tag to be concatenated to the cookie. (See also claims 66 and 86.) Wallace fails to teach or suggest the use of a server identifying tag. Wallace does teach the use of a cookie but fails to teach the use of a tag in addition to the cookie. As defined in the *Dictionary of Computer and Internet Terms*, a "cookie" is "information stored on a user's computer by a WEB BROWSER at the request of software at a web site. *Web sites use cookies to recognize users*

who have previously visited them." Thus, cookie's do not recognize information servers but visitors to the information servers.

This conclusion is supported by Wallace, Jr. Wallace, Jr., states as follows:

> [0004] To remedy the problem of HTTP's stateless nature, cookies have been introduced for the specific purpose of creating states. They may be temporary, in which case they are stored only in memory; or persistent, in which case they are stored in a file, typically on a hard drive, for period of time measured by an expiration date field of a cookie. A cookie may be thought of as a data structure stored in the memory or on the storage device of a user's computer, with the cookie containing data, *such as the user's identity and/or other information about the user* for the purpose of creating a state between the web server and the user. *Thus, when a user visits a particular website, a cookie stored on a user's computer may be sent from the user's computer over the Internet to the web server, which then extracts the data from the cookie, processes the data and therewith creates a state. For example, a user's name may be stored in a cookie and when that user visits a particular website, the data contained in the cookie may be sent to the server and used to identify the user.*
>
> [0006] However, the use of cookies has created a significant problem relating to user privacy. Because these cookies are stored on a user's computer, especially when on a hard drive, other servers may potentially access the cookies of other servers and extract and read the user's identity and/or other information about the user that is stored in those cookies. Such extracting and reading is considered by many as an invasion of the user's privacy.
>
> [0143] The optional parameter (4), the path the cookie is valid for, may be explicitly assigned a value, such as "/computerstore". *This causes to be set the URL path the cookie is valid within. Thus, pages outside the path "/computerstore" cannot read or use the cookie having this value.* Explicitly assigning a value to this parameter would be advantageous where multiple websites exist within a domain, such as www.thissite.com and www.thissite.com/otherparty, and sharing of cookies between the servers associated therewith is undesired. If not specified, the value defaults to the path of the document creating the cookie. The optional parameter (5), the domain the cookie is valid for, may be explicitly assigned a value, such as ".thissite.com". Where a website uses multiple servers within a domain, it may be desirable to make the cookie accessible to pages on any of those servers. Thus, a cookie may be assigned to an individual server or to an entire Internet domain. Here, all servers within the domain www.thissite.com may access

> the cookie so defined. The default value if not explicitly set is the full domain of the document creating the cookie.

(Emphasis supplied.)

The foregoing language makes clear that the cookie identifies the client or the user thereof not the server and that Wallace, Jr., is intended to limit the accessibility of cookies to servers using specific paths to protect the privacy of the identified client user. For at least this reason, the mere teaching of a cookie fails to render obvious the use of a server-specific tag to identify a server and provide fast switching of request packets to the appropriate server.

The Examiner, in response, alleges that Wallace and Balijepalli teach the use of a server identifying tag *in addition* to the cookie; however, the Examiner's response at pages 5-6 of the Office Action uses a client identifying cookie disclosed in Wallace as support. The Examiner seems to be taking the position that, because a cookie is used to identify a client, it would be obvious to use a server tag to identify a server. This is nothing more than hindsight obviousness.

Dependent claim 53 states that, during a first time interval, the flow switch is in a tagging mode in which the switch generates and appends tags to transaction responses and, during a second different time interval, the switch operates in a digesting mode in which digests are generated, invariant hotness is monitored, and transaction requests are routed to information servers based on requested invariant hotness and/or cookie. At col. 5, lines 61-63; col. 5, line 66 to col. 6, line 2; col. 6, lines 54-64; and col. 7, lines 7-15, Jordan, et al., teaches only that the load monitor 120 does not consider the "hotness" of the requested information when no load imbalance condition or loading trend is found to exist. Jordan, et al., fails to teach the use of separate tagging and digesting modes. (See also claims 67 and 87.) Jordan, et al., fails to teach

the use of transaction response tagging, let alone the performance of tagging or hotness monitoring in one mode but not in another mode.

Based upon the foregoing, Applicants believe that all pending claims are in condition for allowance and such disposition is respectfully requested. In the event that a telephone conversation would further prosecution and/or expedite allowance, the Examiner is invited to contact the undersigned.

Respectfully submitted,

SHERIDAN ROSS P.C.

By: Douglas Swartz

Douglas W. Swartz
Registration No. 37,739
1560 Broadway, Suite 1200
Denver, Colorado 80202-5141
(303) 863-9700

Date: Oct 2, 2006